

# LINUX

## 1. Einleitung

1991 von Linus Torvalds erstmalig für den 80386 entwickelt.

Erfolg durch GNU-GPL (Details unter: <http://www.gnu.org/copyleft/gpl.html>).

Heute für viele HW-Plattformen verfügbar: x86, PPC, Alpha, 390, ...

Ein brauchbarer Start für Informationen über Linux: <http://loosgroov.tripod.com/links.html>.

Wesentliche Unterschiede zu den Windows-Systemen:

- Die Betriebssystemlizenz ist frei
- Die gesamte Konfiguration erfolgt über Textdateien
- Der Sourcecode ist verfügbar
- Die graphische Oberfläche ist netzwerkfähig

## 2. Distributionen

Generell ist Linux frei zu verwenden, allerdings ist es relativ aufwendig alle notwendigen Teile aus dem Internet zu kopieren, daher gibt es von einigen Firmen Zusammenstellungen der wichtigsten Teile und meist noch eine Reihe von Anwendungen dazu. Jede Firma entscheidet natürlich für sich, welche Zusätze und Erweiterungen aufgenommen werden, außerdem gilt die GPL nicht unbedingt für alle Teile der Distribution (Lizenz beachten!). Die wichtigsten Distributionen (ohne Anspruch auf Vollständigkeit):

- Red Hat (vor allem in den USA verbreitet)
- S.u.S.E. (in Europa die verbreitetste Distribution)
- Caldera (sehr lange schon graphische Installation, Netwaresupport)
- Corel (relativ neu am Markt)
- Mandrake
- Debian
- Slackware (die klassische Variante; Linux pur)
- ...

## 3. Einsatzgebiete

Netzwerkserver (Fileserver, Applicationserver, Internetserver (Web, DNS, Mail, ...))

Workstation (heute meist mit X-Oberfläche und entsprechender graphischer Shell: KDE, Gnome, ...)

## 4. Generelle Unix-Eigenschaften

- Unix ist ein interaktives Multiuser-/Multitasking Betriebssystem
- Unix ist fileorientiert (d.h. z.B.: jedes Gerät ist eine Datei im Filesystem)
- Unix ist netzwerkfähig

- Unix ist ein offenes Betriebssystem und das einzige für das ein Standard geschaffen wurde (POSIX)
- Unix ist flexibel an die Anwenderbedürfnisse anpassbar (verschiedene Shells, ...).
- Unix ist „relativ“ leicht konfigurierbar und kann den Zugriff auf alle Ressourcen sehr fein einstellen (Nicht nur Benutzer/Administrator).

## 5. Betriebssystemarchitektur

Details siehe z.B.: <http://www.math.uwaterloo.ca/~mctanuan/cs746g/LinuxCA.html>

Shell/Anwendung					Usermode
System-APIs (System Call Interface)					Kernel
Virtual File System (VFS)	character devices	Abstract Network Services	Memory Manager	Process Manager	
FS-Drivers		TCP/IP Driver	VM Driver		
HW-Drivers	HW-Drivers	NIC Driver	Memory Driver	CPU Driver	
HD/FD/CD/...	Konsole/...	NIC	Memory	CPU	Hardware

## 6. Kernel

Der offizielle Linuxkernel wird noch immer von Linus Torvalds mit einer relativ kleinen Gruppe von Programmieren weiterentwickelt, allerdings ist durch den freien Sourcecode niemand an diese offizielle Version gebunden, sondern kann für ihre/seine Bedürfnisse einen maßgeschneiderten Kernel anfertigen (sofern sie/er über genügend Personjahre verfügen kann).

## 7. Filesystems

Hier stehen eine breite Palette von Dateisystemen zur Verfügung, insbesondere alle PC-typischen (FAT, HPFS, NTFS, Minix, CDFS, VFAT, HFS(Apple), ...). Linux bietet auch einige eigene Dateisysteme (extfs2, Reiser, ...) und unterstützt Netzwerkdateisysteme (NFS, DFS, ...) sowie Kryptographische Dateisysteme (TCFS, ...). Ein wesentlicher Unterschied zu DOS/Windows-Systemen ist allerdings der Zugang für den Anwender, da unter Unix-Systemen nur ein logisches Dateisystem existiert, in das alle Geräte (Laufwerke) eingebunden (mount, umount) werden. Dieses logische Dateisystem besteht aus dem Root-Node (/) und darunter liegenden Verzeichnissen, hinter denen sich auch andere Laufwerke oder Teile eines anderen Computers verbergen können, dabei ist es irrelevant mit welchen physischen Dateisystemen die einzelnen Unterverzeichnisse aufgebaut sind.

z.B.:

/	(3. Partition (ext2) auf erster Platte (C:))
/etc	(Unterverzeichnis auf 3. Partition auf erster Platte)
/boot	(2. Partition (ext2) auf erster Platte)
/var	(Unterverzeichnis auf 3. Partition auf erster Platte)
/proc	(Virtuelles Dateisystem zur Verwaltung)
/home	(Unterverzeichnis auf 3. Partition auf erster Platte)

/mnt (1. Partition auf erster Platte mit Dateisystem NTFS)  
/floppy (1. Diskettenlaufwerk (A:) mit Dateisystem FAT)

## 8. Shells

Unter Unix kann der Administrator für jeden Anwender festlegen, mit welcher Shell der Anwender „empfangen“ wird. Das können entweder ein Reihe von Kommandozeilenorientierten Shells sein oder eine X-Oberfläche. Die wichtigsten Shells:

- Bourne-Shell (sh)
- Korn-Shell (ksh)
- C-Shell (csh)
- Bourne Again Shell (bash) - Heute der Quasistandard unter Linux

Alle Shells verstehen die gleiche Syntax und sind mit leistungsfähigen Scriptsprachen ausgestattet, haben aber Unterschiede in ihren Möglichkeiten. Durch diese Scripts sind z.B. automatische Auswertungen von Logfiles möglich, die auf anderen Betriebssystemen nur mit eigens dafür geschriebener Software durchgeführt werden könnten. Die wichtigsten Befehle:

- ls Inhaltsverzeichnis (list)
- cat datei Anzeige von datei
- man befehl Manual für befehl ausgeben
- cd verz Wechsle das Verzeichnis auf verz
- pwd Anzeige des aktuellen Verzeichnisses (print working directory)
- mkdir verz Anlegen eines Verzeichnisses
- rmdir Löscht ein Verzeichnis
- mv alt neu Umbenennen bzw. Verschieben von alt nach neu
- rm datei Lösche datei
- cp quelle ziel Kopiert quelle nach ziel
- chmod mode file Ändern der Berechtigungen
- ...

## 9. X

X ist ein umfangreiches und vor allem netzwerkfähiges System für alle graphischen Anwendungen und heute oft die Standardoberfläche um die unbedarften BenutzerInnen nicht mit einer Kommandozeile abzuschrecken. Die Bedienung und das Aussehen unterscheiden sich auf einem Einzelplatz-PC nicht sehr von einer Win32-Oberfläche, in einem Netzwerk werden aber schnell die Unterschiede deutlich.

Die Komponenten von X:

- X-Server (Steuert die Hardware)
- X-Windowmanager (Graphische Präsentation, Aussehen der Fenster)
- X-Anwendungen (Clientprogramm für die eigentliche Aufgabe)

Unter X ist es möglich einen relativ leistungsschwachen PC mit einem X-Server und einem Windowmanager auszustatten und auf einem anderen Rechner laufende Anwendungen so zu bedienen, als würde diese auf dem leistungsschwachen PC laufen (Senkung der TCO möglich).